Validation of Real-Time Data Processing for the Ground and Air-MSPI Systems

Thomas Werne
Dmitriy L. Bekker
Paula J. Pingree
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, California, 91109-8099
[First.MI.Last]@jpl.nasa.gov

Abstract— JPL is currently developing the multi-angle spectro-polarimetric imager (MSPI), targeted for the Aerosol-Cloud-Ecosystems (ACE) mission, as defined in the National Academies 2007 Decadal Survey. In preparation for the space-based instrument, the MSPI team has built two incremental camera systems (Ground- and Air-MSPI) to gain improved understanding of the proposed architecture. Ground-MSPI is a gimballed instrument used primarily for stationary observation and characterization of the imager and optics. The ER-2 based Air-MSPI operates in a step-and-stare mode, providing multi-angle imaging of a static target. This mode-of-operation simulates the observation scenario of the final space instrument.

Physically, MSPI is a pushbroom camera with a specialized frontend. Before imaging, light entering the camera passes through a pair of photoelastic modulators and a set of pattern polarizers. These optical elements act on the light to make polarimetric extraction computationally feasible. Calculating polarimetric parameters from the imager's data stream requires a real-time least-squares computation that produces coefficients of a truncated time-series expansion of the image. As reported in [1][2], the data processing algorithm can operate in real-time on a Xilinx Virtex-5 FPGA. Moving beyond verification with an onboard data source, the algorithm has been validated on a commercial development board interfaced with the ground camera. In addition, the algorithm has been instantiated within the Air-MSPI electronics board's FPGA, and in situ first-light has been achieved.

TABLE OF CONTENTS

1 INTRODUCTION 1
2 MSPI BACKGROUND 2
3 GROUND-MSPI DEMO SYSTEM 3
4 GROUND-MSPI VALIDATION RESULTS 4
5 AIR-MSPI DEMO SYSTEM 5
6 AIR-MSPI ON-BOARD PROCESSING STATUS 6
7 FUTURE WORK 6
8 CONCLUSION 7
ACKNOWLEDGEMENTS 7
REFERENCES 7

1. INTRODUCTION

The Earth Sciences Decadal Survey identifies a multiangle, multispectral, high-accuracy polarization imager as one requirement for the Aerosol-Cloud-Ecosystem (ACE) mission. JPL has been developing a Multiangle SpectroPolarimetric Imager (MSPI) as a candidate to fill this need. A key technology development needed for MSPI is on-board signal processing to calculate polarimetry data as imaged by each of the 9 cameras forming the instrument. With funding from NASA's Advanced Information Systems Technology (AIST) Program, JPL is solving the real-time data processing requirements to demonstrate, for the first time, how signal data at 95 Mbytes/sec over 16-channels for each of the 9 multi-angle cameras in the spaceborne instrument can be reduced on-board to 0.45 Mbytes/sec. This will produce the intensity and polarization data needed to characterize aerosol and cloud microphysical properties. Using the Xilinx Virtex-5 FPGA we have implemented a least squares fitting algorithm that extracts intensity and polarimetric parameters in real-time, thereby substantially reducing the image data volume for spacecraft downlink without loss of science information[1].

The MSPI project consists of three phases: Ground-MSPI, Air-MSPI, and Space-MSPI. Ground-MSPI is a ground-based camera demonstration focused on characterizing the imager optics and performance. Air-MSPI is an updated version of the ground system which has flown on an ER-2 aircraft in Sept. 2010. Lessons learned from the ground- and air-based demonstrations will be used in the design of the satellite-based Space-MSPI instrument. We have developed a modular demonstration system to test and verify the real-time processing of the FPGA-based on-board processing system first with simulated data, then with ground camera input via a Camera Link interface, and finally with the Air-MSPI camera system.

[1] 978-1-4244-7351-9/11/$26.00 ©2011 IEEE.
[2] IEEEAC paper #1303, Version 1, Updated 10/01/2011.

2. MSPI BACKGROUND

MSPI Camera Systems

The Multiangle SpectroPolarimetric Imager (MSPI) is an instrument concept in development at JPL to produce a highly accurate multiangle-multiwavelength polarimeter to measure cloud and aerosol properties as called for by the Aerosol-Cloud-Ecosystem (ACE) Tier-2 mission concept in the Decadal Survey. Light-cloud/aerosol interactions affect the polarization properties of the light—this polarimetry can be used as a proxy to estimate aerosol refractive index and particle size variances, which are then used as inputs to climate models.

This instrument has proposed 9 cameras (8-fixed and 1-gimballed), each of which must eventually process a raw video signal rate around 95 Mbytes/sec over 16-20 channels for space flight. A key technology development needed for MSPI is on-board processing to calculate polarimetry as imaged by each of the 9 cameras forming the instrument.

MSPI is being developed [3] as a follow on to the Multi-angle Imaging SpectroRadiometer (MISR) [4], that has been operating since 2000 aboard NASA's Terra satellite. The first MSPI camera system developed is called LabMSPI consisting of a single 660-nm spectral band with channels having 0° and 45° polarizers, and no polarizer. LabMSPI was later upgraded to a multi-band version now known as Ground-MSPI. The spectral bands of the Ground-MSPI camera are 335, 380, 445, 470, 555, 660, 865, 935 nm where the bands at 470, 660 and 865 nm are polarimetric.

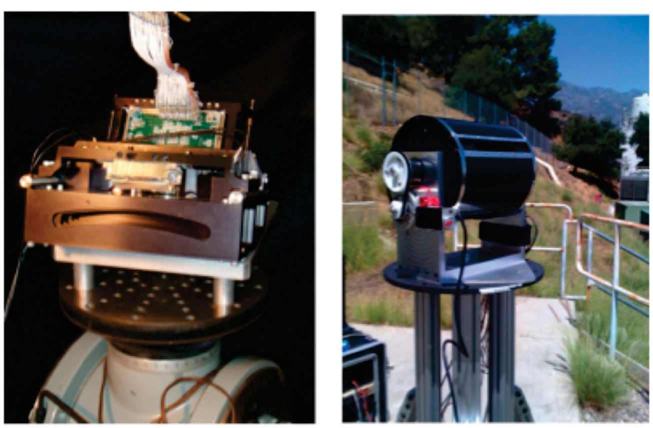

Figure 1. The assembled LabMSPI 660-nm camera (left), and the Ground-MSPI multiband camera deployed outdoors for field experiments at JPL (right).

Air-MSPI is a current NASA Instrument Incubator Program (IIP) development at JPL (D. Diner, PI) which has flown the customized airborne instrument twice in Sept. 2010 on NASA's ER-2 high altitude aircraft. A comparison of MISR and planned MSPI characteristics is summarized in Table 1. The end-to-end real-time data processing results presented in this paper were achieved with the Ground-MSPI camera.

Instrument	Spatial resolution	Along-track angular range	Spectral range (nm)	Polarimetric uncertainty	Field of view
MISR	275 m – 1.1 km	70º fore-70º aft	446-866	NA	±15º
MSPI	125 m – 2.2 km	70º fore-70º aft	355-2130	±0.005	±30º
AirMSPI	27.5 m – 110 m	70º fore-70º aft	355-935	±0.005	±15º
GroundMSPI	30 cm at 1 km distance	Horizon to horizon	355-935	±0.005	±15º
LabMSPI	30 cm at 1 km distance	Horizon to horizon	660	±0.005	±15º

Table 1. Instrument specification comparisons [5]

MSPI Polarization Measurement

In the MSPI design, a dual photoelastic modulator (PEM) assembly is integrated into a polarization-preserving 3-element reflector to provide both intensity and polarization imaging (Figure 2). A miniaturized focal-plane assembly consisting of spectral filters and patterned wire-grid polarizers provides color and polarimetric selection. A custom CMOS array with specialized signal acquisition, readout, and processing electronics captures the radiometric and polarimetric information.

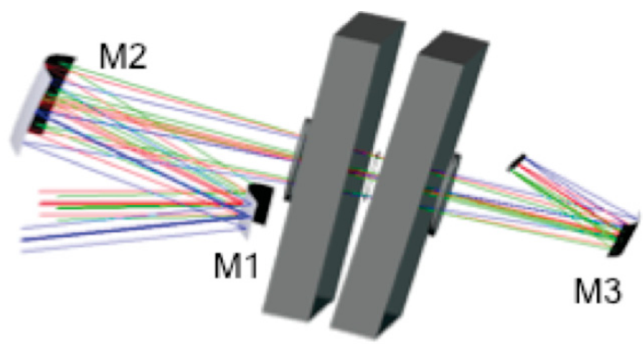

Figure 2. Side view of 3-mirror camera design with integrated dual-PEM [1]

The two photo-elastic modulators (PEMs) are included in the MSPI optical path to modulate the polarization of the transmitted light in order to achieve high accuracy in the degree of linear polarization (DOLP). One full cycle of the modulated polarization signal occurs in the time of one 40-msec frame, set by the difference frequency of the two PEMS. Each cycle of the modulation must be oversampled to create a hi-fidelity digital representation of the polarization components. The baseline is to sample the modulation 32 times per frame thereby creating 32 sub-frames per frame. Compared to the currently operational EOS Mission's MISR cameras, each with 4 spectral channels, the raw video data rate that must be handled by MSPI is increased by a factor of 256 (32x due to oversampling; 4x due to expansion of the number of channels, and 2x due to correlated double sampling to suppress read noise in the Si-CMOS readout).

For a complete mathematical treatment of the MSPI OBP algorithm, see [3][5]. IEEE Aerospace Conference papers from

2009 and 2010 provide a concise overview of the algorithm with an emphasis on implementation details [1][2]. Summary information is provided here so that it can be related to the results presented in Section 4.

One of the goals of the MSPI instrument is to calculate polarimetric information about a target scene. The Stokes vector $\langle I, Q, U, V \rangle$ is used to describe the polarization state of incident light, where I is the intensity, Q and U quantify linear polarization at $0°$ and $45°$, and V quantifies circular polarization. The particular quantities of interest are functions of the Stokes vector components, namely: DOLP, which is defined as:

$$DOLP = \sqrt{(Q/I)^2 + (U/I)^2} = \sqrt{q^2 + u^2}$$

and the associated angle of linear polarization (AOLP), defined as:

$$AOLP = \frac{1}{2} \tan^{-1} (u/q)$$

where the normalized Stokes parameters are $q = Q/I$ and $u = U/I$.

The uncertainty specification on DOLP of $\leq 0.5\%$ is the most challenging aspect of the MSPI design. The MSPI camera measures intensity, I, at several wavelengths in the range shown in Table 1. Q and U are measured in a subset of the bands. The design takes advantage of the fact that the relative measurements q and u can be obtained with higher accuracy than Q or U [3].

MSPI On-Board Processing Implementation

Ground- and Air-MSPI data acquisition is performed with a CameraLink frame grabber. Incoming data is time stamped and stored on an array of parallel hard disks. Ground- and Air-MSPI data reduction via the described algorithm is applied in ground data processing. However, for the Space-MSPI instrument this image data processing must be done on-board the instrument prior to spacecraft downlink due to on-orbit bandwidth constraints to the Earth-based ground stations.

A single 16-channel MSPI camera (one of nine) must process 95 Mbytes/sec of raw video data. A computationally intensive linear least-squares algorithm must also be applied to perform data reduction for video processing of the signal output from the photo-detector array. The result of the on-board processing algorithm is the reduction of dozens of samples acquired during a 40-msec frame to five parameters. Averaging cross-track and along-track pixels to reduce spatial resolution achieves further data reduction.

We have developed an FPGA-based hardware/software co-processor system to implement the algorithm on a Xilinx Virtex-5 FPGA with embedded PowerPC440 processor [1]. In general, the algorithm used to process the polarization channels involves the following steps:

- In hardware, de-multiplex the incoming data stream. Use ancillary information provided about the PEM amplitudes and phases, and time stamps of the subframes within each frame to calculate a set of basis functions.
- In software, create the polarization measurement matrix B, comprised of the sampled basis functions, and calculate its pseudoinverse W.
- Load the W operator into hardware, and then apply it to the sampled measurements using matrix multiplication to retrieve the desired polarization parameter estimates.

The basis functions are analytic expressions (consisting of trigonometric and Bessel functions) of PEM and timing properties: mean retardance amplitude, amplitude difference, the PEM average and difference frequencies, and the integration interval durations and sample locations within the frame.

The implementation of the MSPI OBP algorithm can be thought of as a self-updating multiply-and-accumulate process (which has critical hardware and software components). The processor accepts an input data stream that has been defined by the MSPI electronics and science data teams. The data stream is made up of packets of data; each packet corresponds to a simultaneous sampling of an entire row from the imager. Within each packet are three distinct sets of data: pixel information directly captured from the camera, phase information describing the relative temporal location of the sample within a low frequency PEM oscillation, and ancillary data—including debugging information, state variables describing the PEMs, etc.

3. GROUND-MSPI DEMO SYSTEM

The path to ultimately integrating the MSPI data processing cores and software into the ground-based camera system and eventually the airborne camera system is based on progressively demonstrating important milestones. This ensures that major errors are caught early and leaves room for making design changes based on the performance of incremental builds. Therefore, we have defined a 3-phase development effort. In Phase 1 we generated pseudo-random input data to validate our OBP algorithm design and implementation on the Xilinx ML507 FPGA development board.

Figure 3 shows a block diagram representation of our Phase 2 demonstration system where we replace the pseudo-random input data with an interface to the actual Ground-MSPI camera to validate real-time images. The color key for Figure 3 (and later Figure 9) is: blue indicates hardware/firmware, yellow is software, green borders are complete modules.

Key Development Board Components

The Xilinx ML507 Virtex-5 FXT FPGA Development Board hosts several key components to the Ground-MSPI demo system including the primary design element of the OBP algorithm labeled MSPI Data Processing, the DMA/TCP/Ethernet cores, and the custom software environment called uShell.

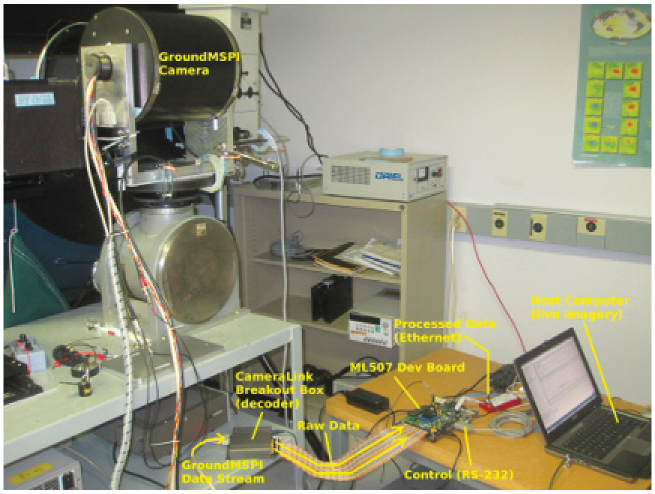

Figure 3. Ground-MSPI Demo System Block Diagram

MSPI Data Processing—The MSPI Data Processing block relies on floating-point matrix multiplication implemented from logic resources on the Virtex-5 FPGA. The V5FXT contains many embedded RAM blocks (BRAM) and dedicated hardware multipliers (DSP48), which are required for this real-time computation [6]. In our design, the logic surrounding BRAM blocks and DSP48 slices is running at 100 MHz, providing a high-speed link between these specialized FPGA resources and the rest of the system.

DMA/TCP/Ethernet—The DMA engine was developed in order to support Ethernet-based real-time data transfer from the FPGA to a host computer. This transfer mechanism was necessary in order to demonstrate an end-to-end integration of the OBP component to the Ground-MSPI camera system. Processed data is sent into a buffer space in main memory via the DMA engine without utilizing the CPU, which is busy with computing parameter updates. Once the buffer space is sufficiently filled, the CPU packetizes the processed data into TCP/IP packets utilizing the Lightweight TCP/IP Stack [7]. The packetized data is then sent to the Ethernet core and put on the Ethernet wire.

uShell—uShell is a minimalist, extensible shell for Xilinx Virtex PPC-class and Microblaze microprocessors that was developed for our demo system, but has wide reaching application to other similar systems [8]. When designing the software portion of the MSPI processor, we wanted to be able to control the device at several levels of granularity. uShell allows a user to call any of a number of parameterizable user-defined functions using a familiar-looking command-line interface. This permits the user to run tests and demonstrations on the platform without having to remember strange key sequences, use terse shortcuts, worry about typing mistakes, or hardcode parameter values. uShell provides a set of useful functions that any design may benefit from (e.g. list devices, memories, and functions, memory and device I/O), plus allows the user to add as many new commands as he/she desires. Plus, since the commands are parameterizable (using the exact same C-based command-line parameter idiom found in Unix systems), the user does not need to write several functions to exercise hardware with different parameter values. In addition, since many hardware peripherals instantiated in FPGAs have reasonably simple register-mapped I/O interfaces, the engineer can edit and view hardware parameter settings at any time without stopping the processor.

Ground-MSPI Camera Interface

Prior to connecting to the Ground-MSPI camera, we worked on validating CameraLink data capture into the Xilinx ML507 FPGA development board. This setup is shown in Figure 4. We used a CameraLink simulator to generate configurable frame patterns at a rate that is specified for the Ground-MSPI camera. CameraLink data was de-serialized using a breakout box and fed directly into the development board (hosting the Virtex-5 FPGA).

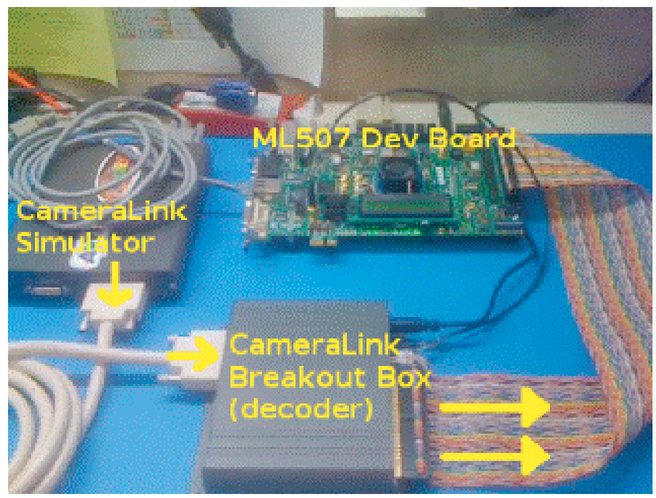

Figure 4. Demo System Hardware (ML507 and CameraLink Decoder)

Once this interface was verified, we substituted the CameraLink simulator with the Ground-MSPI camera (Figure 5). The OBP algorithm running on the FPGA performed real-time data processing of raw imagery from the Ground-MSPI camera and then transferred the derived polarization parameters to the host PC (via Ethernet) for display and plotting using Matlab.

4. GROUND-MSPI VALIDATION RESULTS

These results were recently presented at the Earth Science Technology Forum (ESTF 2010). To demonstrate the functionality of the OBP algorithm operating on the Ground-MSPI camera images, we performed two experiments:

1. Cover a portion of the camera aperture to see the estimated intensity, I, decrease.
2. Rotate a linear polarizer in front of the aperture to see the effect on estimated AOLP.

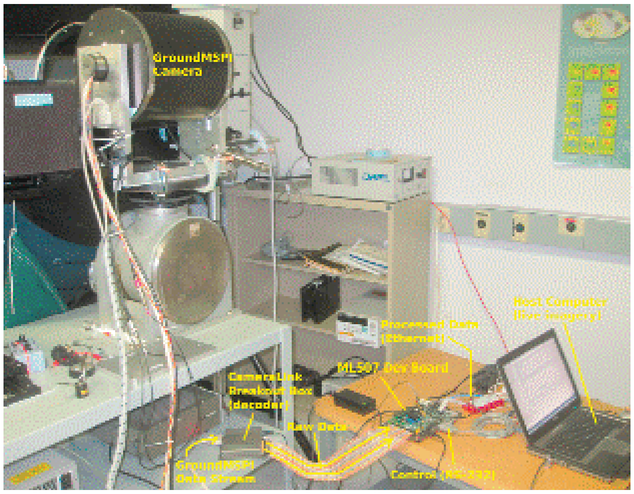

Figure 5. Ground-MSPI Demo System Configuration

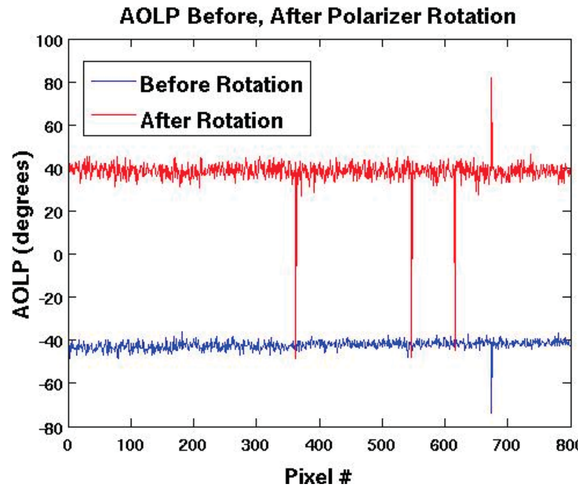

Figure 7. Ground-MSPI AOLP

Figure 6 shows three overlaid snapshots from the first experiment. In this experiment, we began data capture with the aperture unblocked, then covered a portion of the aperture with a block, then passed a narrow obstruction across the entire aperture.

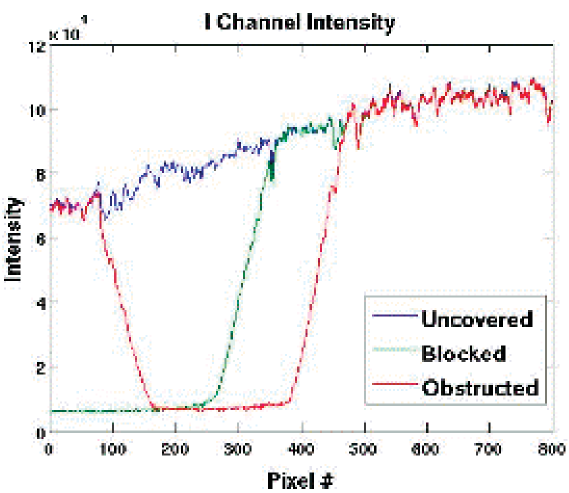

Figure 6. Ground-MSPI Intensity

The first snapshot (blue) is of the unblocked aperture. This shows the nominal camera intensity estimate. Evidently, when the camera images a light source, the image intensity is around 8×10^4. The second image (green) shows the intensity after covering the left side of aperture with a block. Clearly, this is visible in the processed data, since the image intensity drops to about 5×10^3. The third snapshot (red) is with the narrow obstruction in front of the aperture. This also shows the correct behavior, since the intensity is at the blocked level near the obstruction and at the nominal level away from the obstruction. Thus the experimental results agree with the expected results.

Figure 7 shows two snapshots from the second experiment. In this experiment, we began data capture with a linear polarizer between the camera and light source oriented in an arbitrary direction. Midway through the data collection we rotated the polarizer to a different orientation, approximately 90 degrees from the initial state. The blue data shows the computed angle of polarization at the beginning of data collection, and the red data shows the same quantity after rotation. The experimental results match exactly with our expectation.

5. AIR-MSPI DEMO SYSTEM

The Ground-MSPI electronics has a two-board configuration that includes a Focal Plane Assembly (FPA) and a Data Recovery Board (DRB), each with a Spartan-3 FPGA. To demonstrate OBP of the Ground-MSPI images we simply integrated the ML507 development board (with the OBP FPGA algorithm) at the end of the image data stream. The Air-MSPI Electronics has a single Focal Plane Control & Processing (FPCP) board with a military grade Virtex-5FX70T FPGA to enable OBP path-to-flight for ACE (Figure 8). For this demonstration we integrate our OBP algorithm with the rest of the Air-MSPI operational FPGA code for execution on the single device.

For Phase 3 of the demo system, in order to integrate the MSPI OBP algorithm with the Air-MSPI camera system, most of the system level components in the Phase 2 demo are removed (refer to Figure 3). These components are only helpful in validating processed data and are not needed in the final airborne system. We introduce the following elements in Phase 3: FPA Data Acquisition, Data Output, and On-Board CameraLink Encoder IC and Capture Card (Figure 9).

Air-MSPI input data to the MSPI OBP hardware requires a custom data acquisition frontend. We replaced the CameraLink data capture logic from the Ground-MSPI demonstration with direct data capture from the FPA con-

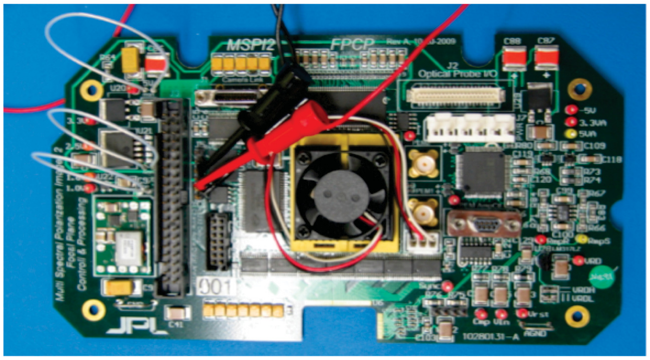

Figure 8. Air-MSPI FPCP Board. The Virtex-5 FPGA is in the center of the board under the fan.

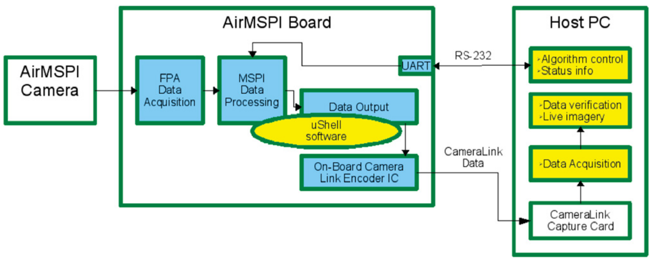

Figure 9. Air-MSPI Demo System Block Diagram

troller core. The processed output data is sent to an off-FPGA on-board CameraLink encoder for streaming to recording units. To implement the data output interface of this demo system we purchased & installed a Camera-Link PCI frame grabber card (EDT PCI DV C-Link). Configuration & data capture is performed using the vendor-provided driver and software API. First a Camera-Link simulator was programmed to output MSPI-format data to successfully capture data using the card with custom software. Then we replaced the Camera-Link simulator with a stand-alone Air-MSPI FPCP board and again captured MSPI-format data using the frame grabber card. Lastly data acquisition software was written to grab and format CameraLink data on the PC.

6. Air-MSPI On-Board Processing Status

In preparation for the ER-2 flight, the FPCP board has been validated in the Air-MSPI camera system for all nominal operations that do not include on-board processing. We have integrated the nominal operation FPGA code with our OBP algorithm on a stand-alone Air-MSPI FPCP board to demonstrate real-time data reduction as will be required in the Space-MSPI system. All that is left is to validate the end-to-end demo system by demonstrating OBP on the FPCP board that is integrated in the Air-MSPI camera system (Figure 10).

Very recently we migrated the OBP algorithm into the FPGA on the Air-MSPI electronics board. After integrating this component with the existing design—which was used to successfully collect raw image data on two ER-2 flights in Sept.

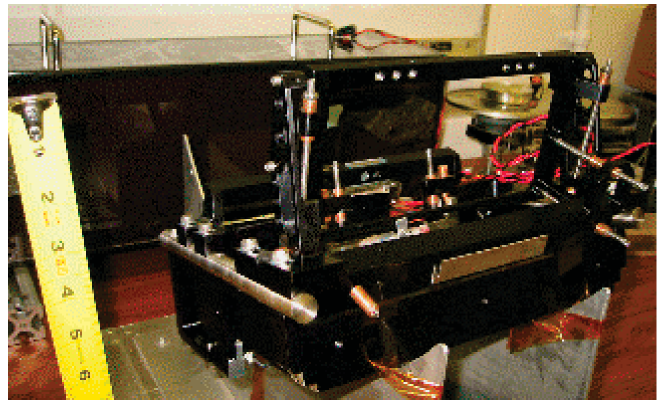

Figure 10. The Air-MSPI Camera

2010—we achieved first-light, as seen in Figure 11. The image shows a time-series (with increasing time downward) readout of processed data. In the middle of the data collection, the aperture of the camera was blocked, then later uncovered. This action can be seen in the data as the appearance and subsequent disappearance of dark bands.

Figure 11. Air-MSPI OBP First Light

The image was captured using software that displays 16-bit integer values as gray-scale intensities. The algorithm's output is a collection of 32-bit floating point numbers, which are then each broken into two 16-bit integers by the frame-grabber software. Although the displayed image has no physical meaning, the distinct change that corresponds exactly with the aperture obstruction shows that the camera electronics and OBP algorithm are operating on real pixel data.

7. Future Work

There are two tasks remaining to finalize the OBP algorithm implementation for MSPI. One is driven by a recent camera design change in pixel-readout timing. The other is due to future considerations for the Space-MSPI camera system where we will use the Xilinx Virtex-5 SIRF[3] device. Xilinx has disabled the PowerPC440 (PPC440) in the SIRF part and so we will replace the current PPC440 implementation with a Microblaze soft-core processor. For Air-MSPI we maintain the PPC440 version of the algorithm.

Mode C Operation

Pixel-readout timing is quasi-static in the Ground-MSPI camera (known operationally as Mode A), but will be changing on a per-frame basis in the Air-MSPI camera. Therefore, we

[3]SIRF = SEU Immune Reconfigurable FPGA

need to alter the current OBP implementation to handle this new mode of operation called Mode C. Figure 12 shows the image frame timing parameters. In Mode A sub-frame sample times are fixed within a frame. In Mode C the sample runs freely.

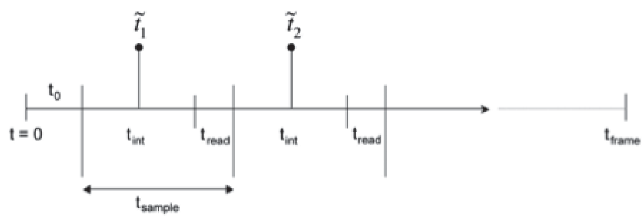

Figure 12. MSPI Frame Timing

The time difference between two sub-frames is constant *within* a frame in Mode A, but not across frame boundaries, whereas it is constant both *within* and *across* frames in Mode C. To implement the change from Mode A to Mode C there are two consequences for the processor. First, the number of sub-frames per frame is variable. Second, sub-frame timestamps are no longer quasi-static such that the basis functions must be updated every frame.

The way we solve this is to create a calibration function that:

1. Samples sub-frame sample times for a frame
2. Computes a set of basis functions
3. After computation, grabs another set of sub-frame sample times and computes the sample time difference
4. Repeat 1-3 and averages the sample time deltas

The idea is that if the instrument is quasi-static, then we can use this approximately fixed average time delta to predict the sub-frame sample times 2-3 frames ahead of the existing sample times, compute the basis functions based on those predicted values, and then load those into the processor.

For Mode A operation, we use the fact that the sample times are fixed. This means that we only need to grab a set of time stamps, compute the basis functions for that set, then load those into the processor. With fixed timestamps, it doesn't matter that we're using "incorrect" timestamps, because the internal control loop guarantees that the drifts are slow relative to our processing timescale.

Microblaze Soft-Core Processor

A MicroBlaze instantiation was selected to replace the PPC440 in the Xilinx Virtex-5 SIRF device because it uses the same buses to interface with logic making it easy for us to port from the PPC440, and it has an optional FPU that is required for pseudo-inverse computation in the OBP algorithm. The key differences between the PPC440 & MicroBlaze are:

- PPC440 is a hard-core; MicroBlaze is a soft-core, taking up FPGA resources
- PPC440 maximum frequency is 550 MHz; MicroBlaze is 250 MHz
- PPC440 has built-in cache support; MicroBlaze uses BRAM resources for cache
- PPC440 is a superscalar processor capable of multiple instructions per cycle; MicroBlaze is single issue

Fortunately, the differences listed above do not pose any challenges for our design.

8. Conclusion

JPL has been developing a Multiangle SpectroPolarimetric Imager (MSPI) to fulfill hardware needs for the Aerosol-Cloud-Ecosystem (ACE) mission. As part of meeting the real-time data processing requirements to demonstrate that signal data at 95 Mbytes/sec over 16-channels for each of the 9 multiangle cameras in the spaceborne instrument can be reduced on-board to 0.45 Mbytes/sec, JPL is working on the Xilinx Virtex-5 FPGA platform. Using this platform, a polarimetric processing least-squares fitting algorithm is under development to meet MSPI's on-board processing requirements. This paper presents the results of on-board processing of Ground-MSPI camera images using the Xilinx Virtex-5 FPGA and current Air-MSPI integration status. We implemented a least-squares fitting algorithm that extracts intensity and polarimetric parameters in real-time, thereby substantially reducing the image data volume for spacecraft downlink without loss of science information. Using our demonstration systems for the Ground-MSPI camera, we performed two validation experiments. The results from these experiments show that the OBP algorithm is processing image data correctly. Additionally, we have acquired first-light for the processing algorithm running onboard the Air-MSPI electronics board that shows end-to-end data flow.

Acknowledgments

The authors wish to acknowledge Sven Geier (JPL) for providing the photo of the Air-MSPI camera (Figure 10).

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration, and by the University of Michigan.

References

[1] T. A. Werne, D. L. Bekker, and P. J. Pingree, "Real-Time Data Processing for an Advanced Imaging System Using the Xilinx Virtex-5 FPGA," *Aerospace Conference, 2010 IEEE*, pp. 1–9, March 2010.

[2] C. D. Norton, T. A. Werne, P. J. Pingree, and S. Geier, "An Evaluation of the Xilinx Virtex-4 FPGA for On-Board Processing in an Advanced Imaging System," *Aerospace Conference, 2009 IEEE*, pp. 1–9, March 2009.

[3] D. J. Diner, A. Davis, B. Hancock, G. Gutt, R. A. Chip-

man, and B. Cairns, “Dual-photoelastic-modulator-based polarimetric concept for aerosol remote sensing,” *Applied Optics*, vol. 46, no. 35, pp. 8428–45, 2007.

[4] D. Diner, J. Beckert, T. Reilly, C. Bruegge, J. Conel, R. Kahn, J. Martonchik, T. Ackerman, R. Davies, S. Gerstl, H. Gordon, J.-P. Muller, R. Myneni, P. Sellers, B. Pinty, and M. Verstraete, “Multi-angle Imaging SpectroRadiometer (MISR) instrument description and experiment overview,” *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 36, no. 4, pp. 1072–1087, July 1998.

[5] D. J. Diner, A. Davis, B. Hancock, S. Geier, B. Rheingans, V. Jovanovic, M. Bull, D. Rider, R. A. Chipman, A.-B. Mahler, and S. C. McClain, “First results from a dual photoelastic modulator-based polarimetric camera,” *Applied Optics*, pp. 2929–2946, 2010.

[6] *Embedded Processor Block Reference Guide (UG200)*, Xilinx, Inc., San Jose, CA, January 2009.

[7] A. Dunkels, “The lwIP TCP/IP stack,” http://www.sics.se/~ adam/lwip/documentation.html, Swedish Institute of Computer Science, 2004.

[8] T. A. Werne, “uShell — A Minimalist Shell for the Xilinx Virtex PPC-Class and Microblaze Microprocessors,” *New Technology report (NTR-47495)*, April 2010.

Thomas A. Werne is an Electronics Engineer II in the Instrument Flight and GSE Software group in the Instrument Software and Science Data Systems Section at JPL. He is currently working on implementing FPGA-based technology for Smart Payload applications. Thomas has Bachelor of Science degrees in Electrical Engineering and Mathematics, a Master of Electrical and Computer Engineering from Rose-Hulman Institute of Technology in Terre Haute, IN, and is currently enrolled as a Ph.D. Student in Control and Dynamical Systems at Caltech. He is a member of IEEE.

Dmitriy L. Bekker is an Electronics Engineer II in the Instrument Flight and GSE Software group in the Instrument Software and Science Data Systems Section at JPL. He is currently working on FPGA-based data acquisition and processing design for multiple prototype science instruments. His areas of interests include FPGAs, embedded systems, digital signal processing, and system architecture. Dmitriy received his M.S. and B.S. degrees in Computer Engineering from Rochester Institute of Technology in 2007.

Paula J. Pingree is Supervisor of the Flight Instrument Electronics group in the Instruments and Science Data Systems Division at JPL, and Principal Investigator for the AIST “Optimizing MSPI Design for the ACE Mission” technology task. She has been a key contributor to the design, integration, test and operation of several JPL flight projects including Cassini, Mars Global Surveyor, Deep Space 1, and Deep Impact. Most recently she led the Electronics development for the Microwave Radiometer instrument on the Juno spacecraft planned for a 2011 launch to Jupiter. Ms. Pingree has a Bachelor of Engineering degree in Electrical Engineering from Stevens Institute of Technology in Hoboken, NJ, and a Master of Science in Electrical Engineering from California State University Northridge. She is a member of IEEE.